⑫ EUROPEAN PATENT APPLICATION

㉒ Inventor : Suzuki, Naoshi
3-1-2-916 Kyomachi
Kawasaki-ku, Kanagawa-ken (JP)
Inventor : Uno, Shunya
Sunwhite M401-13, 4-8-7 Minaminaruse
Machida-shi, Tokyo (JP)

㉔ Representative : Killgren, Neil Arthur
IBM United Kingdom Limited Intellectual
Property Department Hursley Park
Winchester Hampshire SO21 2JN (GB)

�554 **Reducing power consumption in a digital processor.**

�557 The supply of the clock signal to the processor in a multi-tasking system is controlled by a program (CCP) that runs in the lowest priority under the operating system (OS), so that applications (A,B,) are not affected when clock signals are stopped or slowed to reduce power dissipation.

FIG. 1
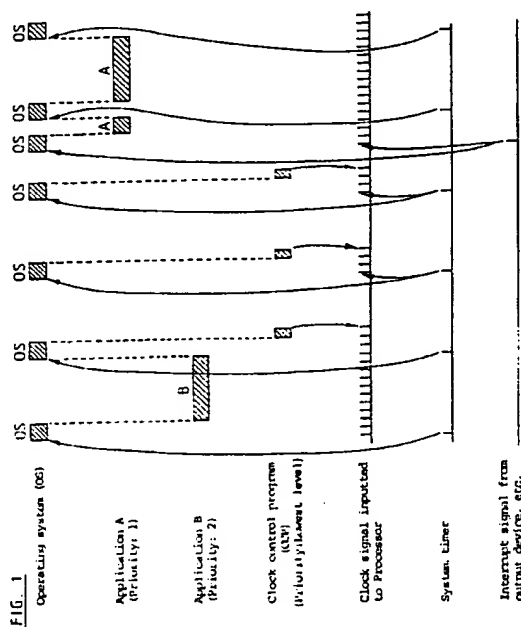
EP 0 501 655 A2

The present invention relates to reducing the power consumption of a digital processor. Such reduction is desirable particularly to reduce heat dissipation in data processing apparatus including digital processors.

In a known type of processor the contents of its internal registers are not lost even if the supply of a clock signal to the processor is stopped. Operations can be resumed from the stop state by reopening the supply of the clock signal. Such a processor is referred to herein as a full-static processor and, if it uses CMOS logic, power consumption and heat dissipation can be greatly reduced by stopping the supply of a clock signal to the processor, or decreasing the frequency of the clock signal, while the processor idle.

Japanese Published Unexamined Patent Application (PUPA) No.62-169219 (U.S. Pat. No.4,851,987) discloses an information processing system in which the supply of a clock signal to a processor is stopped according to a result of the execution of a program which determines whether the system is currently in a state where it waits for the completion of an operation of an input/output device or key input by an operator. However, even though the system waits for the completion of the operation of the input/output device or key input by the operator, the processor is not always used to be in an idle time. Particularly, in an information processing system using a multi-tasking operating system, possibility that a processor runs a second task while waiting for the completion of an operation of an input/output device or key input by an operator is not insignificant. In such a conventional apparatus there is therefore a danger that the supply of a clock signal to the processor may be stopped while the processor runs the task.

It is accordingly an object of the present invention to provide an information processing system and a method for operating such a system which minimises this danger.

According to the present invention there is provided an information processing apparatus using a multi-tasking operating system, a method for controlling a processor clock signal wherein the supply of a processor clock signal to a processor is controlled by a program which runs in the lowest priority under said multi-tasking operating system.

There is further provided an information processing system including a full static processor, a clock signal generator for supplying the clock signal to said processor, and a clock signal controller for controlling the supply of the clock signal to said processor from said clock signal generator when a program given the lowest priority runs under a multi-tasking operating system.

An information processing system is usually provided with a system timer for synchronising the entire system in addition to a generator or oscillator for generating the clock signal to be supplied to the processor. Under the multi-tasking operating system, tasks to run are switched to one another according to a priority given to each task for each time interval indicated by the system timer. Now in a case where a program (clock control program) that controls the supply of the clock signal to the processor is provided, and the clock control program is given the lowest priority, the clock control program runs if only the clock control program runs and any other tasks does not run, that is, only if the processor may be stopped. Thereby, the supply of the clock signal to the processor can be certainly controlled on or off for each time interval indicated by the system timer. Instead of stopping the supply of the clock signal to the processor, the frequency of the clock signal may be controlled so that it can be decreased.

In order that the invention may be well understood a preferred embodiment thereof will now be described by reference to the accompanying drawings, in which:-

FIG.1 is a time chart showing the operations of the embodiment of an information processing system constructed in accordance with the present invention.

FIG.2 is a block diagram showing the overall construction of said embodiment.

FIG.3 is a flowchart showing a method for controlling processor clock signals and processing steps according to the present invention.

FIG.2 shows the overall construction of an embodiment of an information processing system according to the present invention. Referring to the figure, a processor 10 connects to a clock signal line 21, an interrupt signal line 22, and a signal lines 23. The signal line 23 is comprised of a plurality of signal lines for a control signal, an address signal, and a data signal other than an interrupt signal and connected to a memory 30 and a system timer 70. The signal line 23 and the interrupt signal line 22 may be called a system bus 20. The memory 30 stores a multi-tasking operating system MOS, a clock control program CCP that runs in the lowest priority under the multi-tasking operating system MOS, and application programs A and B.

The clock signal line 21 and the interrupt signal line 22 connect to a clock signal controller 40. The clock signal controller 40 comprises a clock signal on-off switcher 41, a signal transition detector 42, and a register 43. The clock signal on-off switcher 41 is always supplied with a clock signal for the processor from a generator 50, stopping the supply of the clock signal to the processor 10 in response to a clock stopping signal from the register 43, reopening the supply of the clock signal to the processor 10 in response to a clock reopening signal from the signal transition detector 42. The signal transition detector 42 connects to an interrupt signal controller 60. The signal transition detector 42, when receiving an interrupt signal from the interrupt signal controller 60, gives the

interrupt signal to the processor 10 and the clock reopening signal to the clock signal on-off switcher 41.

To the interrupt signal controller 60, in addition to interrupt signals from input/output devices such as a keyboard, a disk storage, a printer, etc., a signal from the system timer are inputted. When receiving these signals, the interrupt signal controller 60 outputs the interrupt signal to the signal transition detector 42. Thereby, except that the interrupt signals from the input/output devices are received, the supply of the clock signal to the processor 10 is reopened when the system timer 70 indicates the lapse of a predetermined time. Under the multi-tasking operating system MOS, task execution is switched for each time interval indicated by the system timer 70. The register 43 is used for receiving and holding a result of the execution of the clock control program CCP given the lowest priority.

Now the operations of the embodiment are described by reference to FIG.1 in addition to FIG.2. Generally, in the multi-tasking operating system, a task scheduler, which is one of the component programs of the operating system, switches tasks on or off based on time slicing in which the task scheduler gives plural tasks the execute right of the processor according to their priority for each predetermined time interval at the completion of a running task or based on a event-driven method in which the execute right of the processor is transferred in response to occurrence of an event such as an interruption, as in the embodiment. Time quanta of time slices are determined by the system timer 70.

FIG.1 shows a state where the execution of plural tasks including the clock control program CCP given the lowest priority is switched under the multi-tasking operating system OS and a condition that the supply of a clock signal to the processor 10 is controlled on or off. Now if the system timer 70 issues a signal indicating the lapse of a predetermined time interval over the signal line 23 while the application program B given priority 2 is running, an object of program execution temporarily changes from the application program B to the operating system OS. The operating system OS not only determines whether the application program B has the execute right of the processor 10 even in the following quantum of time, but also selects any of programs that require the execute right of the processor 10 in the following quantum of time and gives it the execution right.

FIG.1 shows a case where only the clock control program CCP requires the execute right of the processor 10. In this case, the clock control program CCP runs and, as a result, a signal indicating that a clock signal may be stopped is sent to the register 43 of the clock controller 40 through the signal line 23. Since the clock control program CCP does nothing but simple work as described above, its execution time is very short. The register 43 not only holds said signal, but

also provides a clock stopping signal to the clock signal on-off switcher 41 to stop the supply of the clock signal to the processor 10.

If the supply of the clock signal to the processor 10 stops and then the system timer 70 issues a signal indicating the lapse of a predetermined time, the interrupt signal controller 60 provides an interrupt signal to the signal transition detector 42 and then the signal transition detector 42 provides a clock reopening signal to the clock signal on-off switcher 41 to not only reopen the supply of the clock signal to the processor 10, but also provide an interrupt signal to the processor 10. The processor 10, on receiving the interrupt signal, prompts a timer interrupt handler and a dispatcher, which are the component programs of the operating system OS, to run and determines a program to be run in the following quantum of time after they run.

FIG.1 shows also a case where a program to be run in the following quantum of time is only the clock control program CCP. Also in this case, the clock control program CCP runs to stop the supply of a clock signal to the processor 10. If the interrupt signal controller 60 issues an interrupt signal while the supply of the clock signal to the processor 10 is stopping, not only the supply of the clock signal to the processor 10 is reopened, but also the operating system OS performs functions to determine a program to be run in the following quantum of time. If the program to be run is the application program A given priority 1, the application program A runs until the system timer 70 issues the following signal. When the system timer 70 issues the signal, the operating system functions to determine a program to be run in the following quantum of time and if the program thus determined is the application program A, the program A runs again until the system timer 70 issues the following signal.

FIG.3 shows processing steps for stopping the supply of a clock signal to the processor under the multi-tasking operating system. In the figure, both an interrupt handler and a task dispatcher are the component programs of a task scheduler of the operating system. The interrupt handler is a program that runs in response to an interrupt signal to do work required for stopping and then reopening an application running before receiving the interrupt signal. The task dispatcher is a program that determines the following task to be run according to priority given to each task and transfers the execute right of the processor to the task thus determined.

Referring to FIG.3, in a step S11 an application program given priority other than the lowest priority runs. In a subsequent step S12, whether said application program is completed or not is determined. If the application program is not completed, the processing proceeds to a step S13 where the application program continues to run. If the application program is completed, the processing proceeds to a step S16.

If an interrupt signal is encountered in a step S14 during the run of the application program, the processing proceeds to a step S15 where the interrupt handler runs and then the task dispatcher is started in the step S16. The task dispatcher determines whether the running application program is authorised to run also in the subsequent quantum of time or not (in a step S17). If the application program is authorised to run in the subsequent quantum of time, the processing returns to the step S11. If the application program is not authorised to run, whether another application program requests authorisation for running or not is determined in a step S18. If another application requests to be authorised to run, the application program runs in a step S19. Otherwise, the processing proceeds to a step S21 where the clock control program (given the lowest priority) for stopping a clock signal is started. In a subsequent step S22, the clock control program runs to send a signal indicating that the supply of the clock signal is stopped, to the register 43 of the clock signal controller 40 and in a step S23 the supply of the clock signal is stopped.

According to the above embodiment, since the clock control program runs only if there is no other tasks to be run, the supply of the clock signal is not in danger of being stopped while some task is running in a background. Further, according to the embodiment, since the on-off control for the clock signal is performed for each timer interval indicated by the system timer, the clock signal is stopped in a small slice of the idle time for the processor and thereby low power dissipation is offered.

It will be recognised that said clock signal controller 40 may be connected to a processor, not the processor 10 and added to another program. It will be appreciated that instead of stopping the supply of a clock signal to the processor 10, the frequency of the clock signal may be decreased to attempt low power dissipation.

As described above, according to the present invention, a method and an information processing system which ensure that an allowed time to stop a processor or an allowed time to delay an operation of the processor is determined to stop the supply of a clock signal to the processor or decrease the frequency of the clock signal, can be provided.

Claims

1 In an information processing apparatus using a multi-tasking operating system, a method for controlling a processor clock signal wherein the supply of a processor clock signal to a processor is controlled by a program which runs in the lowest priority under said multi-tasking operating system.

2 The method for controlling a processor clock signal according to claim 1 wherein said program is

adapted to open or to close the supply of clock signals to the processor.

3 The method for controlling a processor clock signal according to claim 1 wherein said program is adapted to change the frequency of the clock signal provided to the processor from an operating frequency to a reduced idle frequency and vice versa.

4 An information processing system including a full static processor, a clock signal generator for supplying the clock signal to said processor, and a clock signal controller for controlling the supply of the clock signal to said processor from said clock signal generator when a program given the lowest priority runs under a multi-tasking operating system.
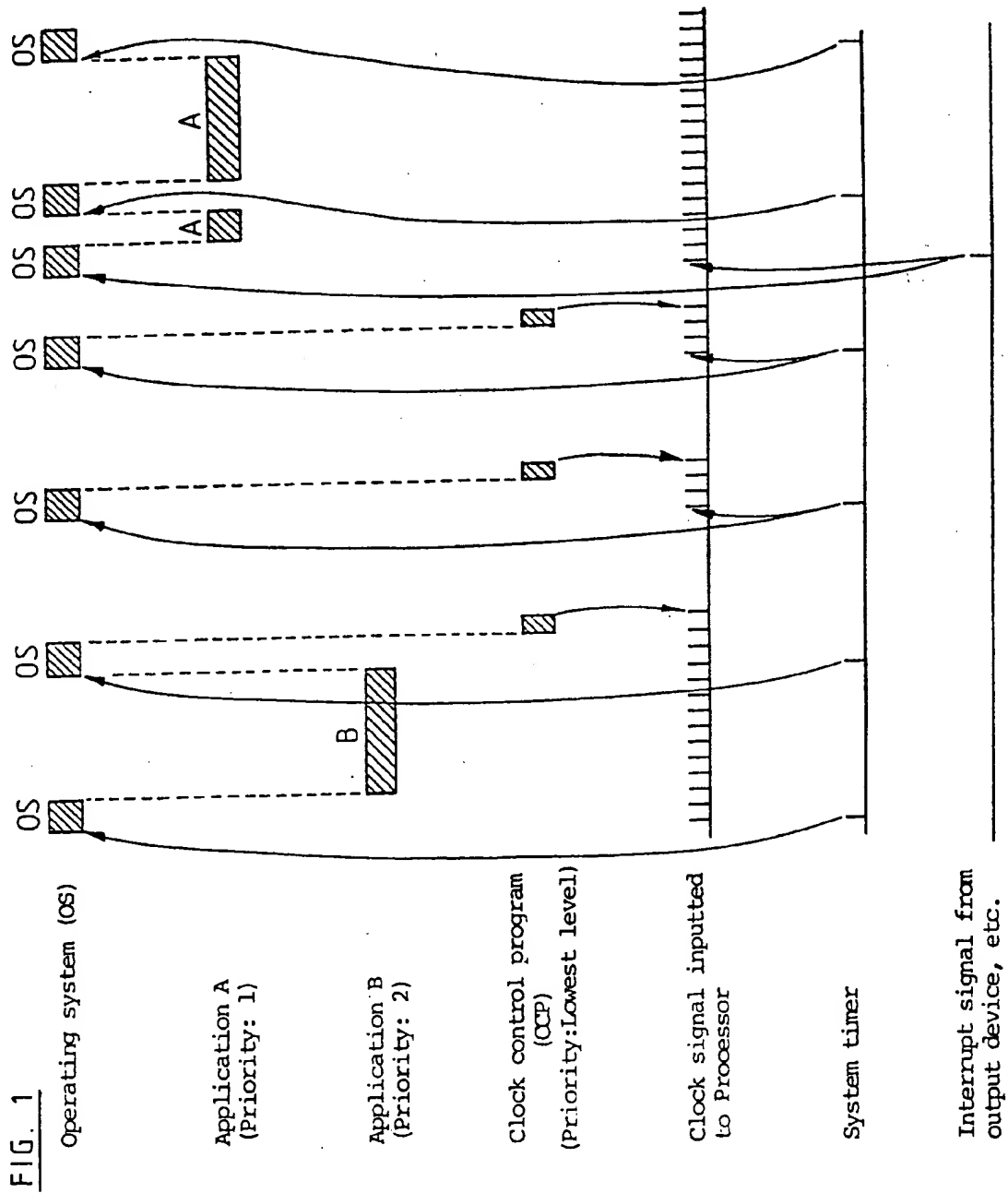
5 A system as claimed in claim 4 in which said clock signal controller is adapted to change the frequency of said clock signal from an operating frequency to an idling frequency and vice versa.

6 A system as claimed in claim 4 in which said clock signal controller is adapted to open or to close the supply of clock signals to said processor.

7 The information processing system according to claim 6 wherein said clock signal controller is adapted to reopen the supply of a clock signal to said processor in response to an interrupt signal to said processor.

8 The information processing system according to claim 7 wherein said interrupt signal is an output signal from a system timer for synchronising the entire information processing system.

9 An information processing system as claimed in any of claims 4 to 8 including a memory, a processor, an input/output device, a generator for supplying a clock signal to said processor, and a system timer for synchronising among the parts of the system, wherein a clock signal controller is provided for the purpose of stopping the supply of the clock signal to the processor by means of a program that runs in the lowest priority under a multi-tasking operating system and reopening the supply of said clock signal in response to an interrupt signal from said input/output device or said system timer.
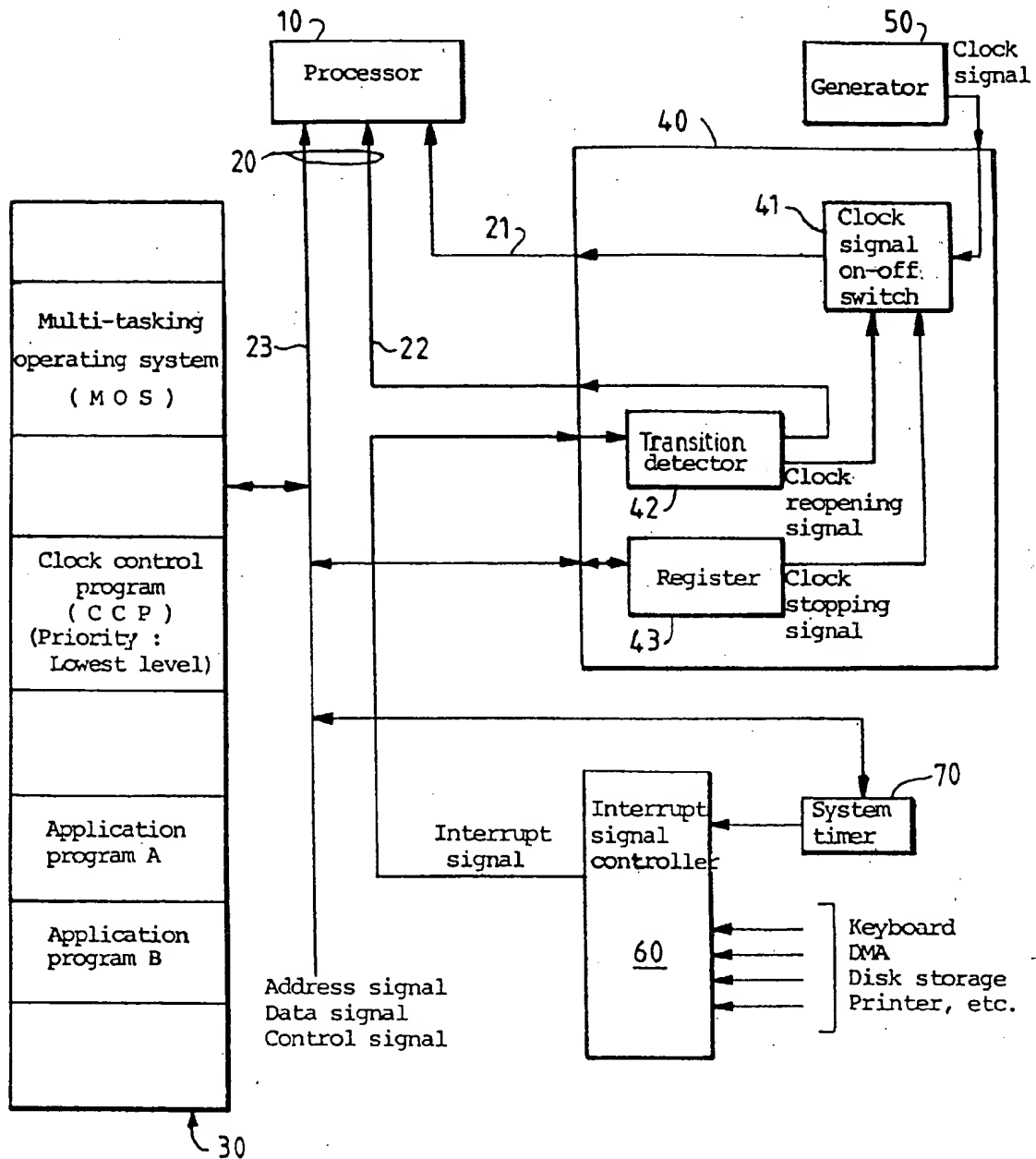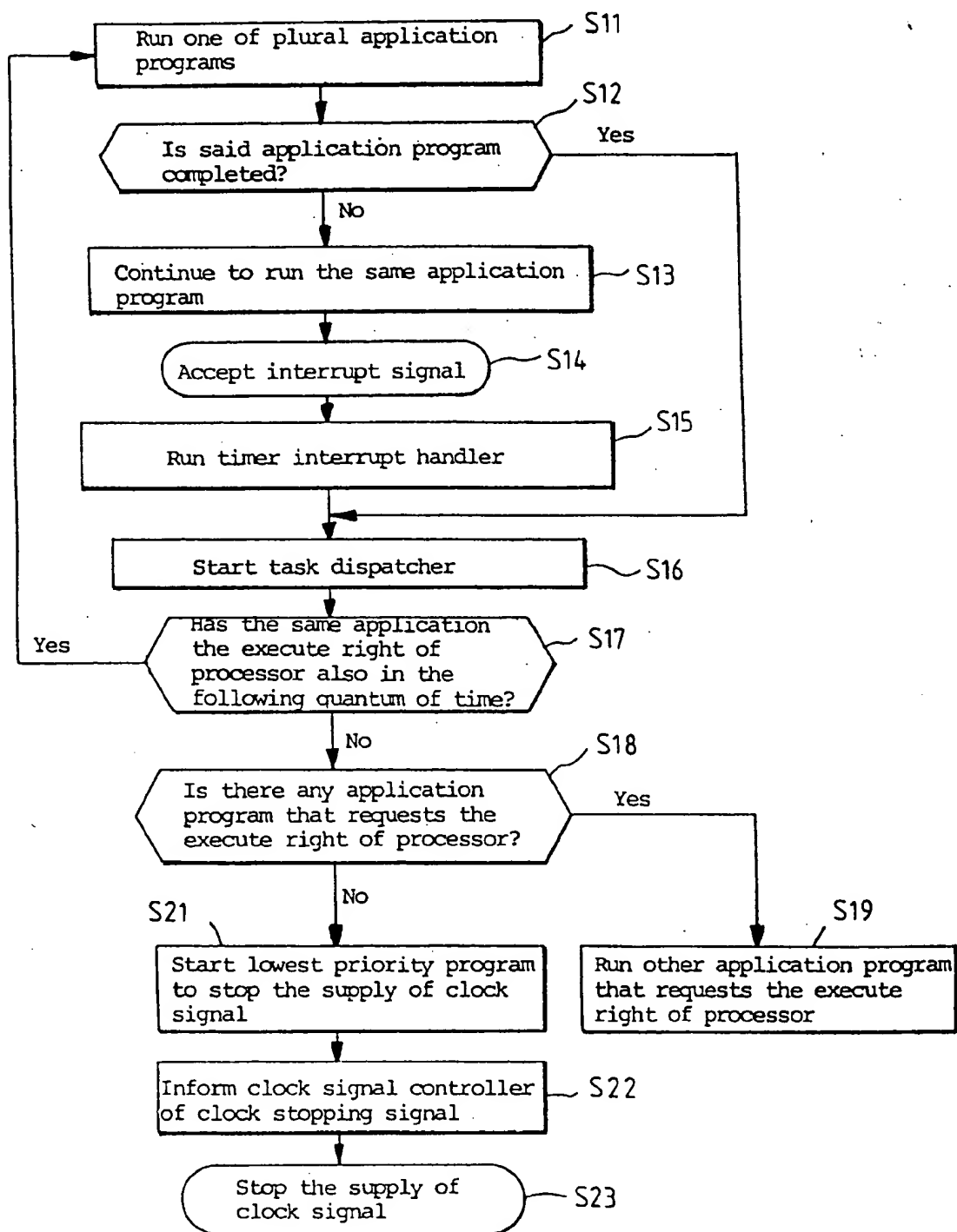
FIG. 1

Operating system (OS)

Application A
(Priority: 1)

Application B
(Priority: 2)

Clock control program
(CCP)
(Priority:Lowest level)

Clock signal inputted
to Processor

System timer

Interrupt signal from
output device, etc.

FIG 2

Run one of plural application programs — S11

Is said application program completed? — S12
Yes

No

Continue to run the same application program — S13

Accept interrupt signal — S14

Run timer interrupt handler — S15

Start task dispatcher — S16

Has the same application the execute right of processor also in the following quantum of time? — S17
Yes

No

Is there any application program that requests the execute right of processor? — S18
Yes

No

S21
Start lowest priority program to stop the supply of clock signal

S19
Run other application program that requests the execute right of processor

Inform clock signal controller of clock stopping signal — S22

Stop the supply of clock signal — S23

FIG. 3